

Atty. Docket No. MS305277.1

DYNAMIC STANDARDIZATION FOR SCORING
LINEAR REGRESSIONS IN DECISION TREES

by

Bo Thiesson and David M. Chickering

MAIL CERTIFICATION

I hereby certify that the attached patent application (along with any other paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on this date July 28, 2003, in an envelope as "Express Mail Post Office to Addressee" Mailing Label Number EV330020378US addressed to the Mail Stop Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.



Himanshu S. Amin

Title: DYNAMIC STANDARDIZATION FOR SCORING LINEAR REGRESSIONS
IN DECISION TREES

TECHNICAL FIELD

5 The present invention relates generally to systems and methods that facilitate data mining and model evaluation. More particularly, the present invention relates to an efficient process for scoring splits in a decision tree in order to evaluate a model's performance while mitigating effects of any parameter prior used to score models.

10

BACKGROUND OF THE INVENTION

 When learning decision trees for continuous variables (or other types), a scoring criterion employed to evaluate how well a tree fits a set of data is often a function of a prior distribution over respective parameters of the tree. In many instances, it is desirable
15 that this prior information have as little effect on the resulting tree as possible. For example, assuming Normal distributed data, when modeling linear regressions in the leaves of a decision tree, a Bayesian scoring criterion generally requires a prior mean on respective inputs (regressors), a prior mean on a target, and a prior covariance matrix over the inputs and the target. Without knowing the domain of the problem, it is often
20 difficult to anticipate what a useful set of priors will be. For instance, data could be in a 0.01 to 0.001 range or in a 100 to 1000 range in which case the prior distributions with least effect are very different.

 One solution to the above problem is to pre-standardize data so that the data has a mean zero and standard deviation of one, thus utilizing a prior mean of zero for all
25 variables and assuming a prior covariance matrix to be diagonal (*i.e.*, assume apriori that all variables are independent). One problem with this solution is that after splitting on a variable in the decision tree, the data that results in different leaf nodes may have very different ranges, and therefore the original problem with the parameter prior is postponed until later in the learning algorithm. It is also not favorable to shift or scale data each time

a new split is considered in the tree, as this will generally cause an enormous runtime performance reduction due to the additional scaling or shifting operations. To illustrate scaling and shifting of data, the following example is provided.

A variable x^k can be employed to denote a variable in some domain. A variable \bar{x}_i denotes a vector of values for a set of variables in the i th case in the data. A variable x_i^k denotes a value of the variable x^k in the i th case. For example, if the data is:

	x^1	x^2	x^3
Case 1	1	4	5
Case 2	9	8	7

Then, $\bar{x}_1 = (1,4,5)$ and $x_2^2 = 8$.

Shifting the n cases $\bar{x}_1, \dots, \bar{x}_n$ is defined as subtracting, for each variable x^k , a mean $m^k = \sum_{i=1}^n x_i^k / n$ from each case.

After shifting the data above yields:

	x^1	x^2	x^3
Case 1	-4	-2	-1
Case 2	4	2	1

Scaling the cases is defined as dividing the value of each variable in each case by the standard deviation for that variable. The standard deviation for a variable is defined as:

$$SD(x^k) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i^k - m^k)^2}$$

(As can be appreciated, there are alternative formulas for standard deviation.)

In this example: $SD(x^1)=4$, $SD(x^2)=2$, $SD(x^3)=1$, and thus the scaled data is

	x^1	x^2	x^3
Case 1	1/4	4/2	5
Case 2	9/4	8/2	7

Generally, standardizing the cases is defined as first shifting and then scaling the data. The result is that each variable will have a mean of zero and a standard deviation of one in the data. As can be appreciated, with larger data sets and number of cases, and as decision trees grow in complexity, standardizing operations can be quite burdensome in terms of system performance such as the large amount of computer computations that may be required to perform such operations.

SUMMARY OF THE INVENTION

The following presents a simplified summary of the invention in order to provide a basic understanding of some aspects of the invention. This summary is not an extensive overview of the invention. It is not intended to identify key/critical elements of the invention or to delineate the scope of the invention. Its sole purpose is to present some concepts of the invention in a simplified form as a prelude to the more detailed description that is presented later.

The present invention relates to systems and methods for automatically evaluating statistical models in a computationally efficient manner. In one aspect, models are processed and learned from continuous variable data or subsets thereof to generate a decision tree that is employed to predict target data for various data types (*e.g.*, economic data, mathematical data, age, education, income, and so forth). Such predictions can be utilized for various data mining applications, for example. In order to determine the accuracy or performance of the models, scores are generated at decision splits or leaves on the decision tree. When scoring a selected split in the decision tree, the present invention utilizes a same (or similar) prior distribution together with non-standardized data - yet scores the split as if the data at respective leaves had been shifted and/or scaled. This mitigates the effect of the prior distribution on (1) the shape of the tree and (2) the distributions in the leaves of that tree, and facilitates superior out-of-sample prediction accuracy.

As noted above, one aspect of the present invention is to score splits in a decision tree as if the data were shifted and/or scaled after the split. Thus, a learning component is provided that inputs continuous variable data and generates a decision tree having one or more decision branches or leaves that are then scored to determine the prediction capabilities of the model. A scoring component automatically analyzes the decision tree in conjunction with various processing modes to determine respective scores for selected branches of the tree. After the branches have been scored, evaluation data can be generated indicating how well a model predicts continuous target data and therefore whether or not the model is a suitable predictor for the target data. The evaluation data can then be employed by users and/or subsequent automated components when determining model performance and/or selecting between models or model subsets.

An optimization component can also be provided with the present invention to calculate scores in the above manner without actually shifting and/or scaling the data. This allows resulting decision trees to be less sensitive to “uninformative” prior information, without incurring the runtime performance penalty that may result from rescaling data each time a new split is considered. Also, actual shifting and/or scaling operations can be applied to the data, if desired. It is noted that although continuous variable data is processed in accordance with the present invention, other types of data can also be processed. For example, since continuous target variables are generally predicted, input data applied to the learning models and resultant decision trees can be continuous data, non-continuous data, data subsets, and/or combinations thereof.

To the accomplishment of the foregoing and related ends, certain illustrative aspects of the invention are described herein in connection with the following description and the annexed drawings. These aspects are indicative of various ways in which the invention may be practiced, all of which are intended to be covered by the present invention. Other advantages and novel features of the invention may become apparent from the following detailed description of the invention when considered in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic block diagram illustrating a model evaluation system in accordance with an aspect of the present invention.

5 Fig. 2 is a diagram illustrating an exemplary scoring component in accordance with an aspect of the present invention.

Fig. 3 is a flow diagram illustrating an example shifting and scaling determination in accordance with an aspect of the present invention.

Fig. 4 is a flow diagram illustrating intermediate scoring processes in accordance with an aspect of the present invention.

10 Fig. 5 is a flow diagram illustrating an example scoring process in accordance with an aspect of the present invention.

Fig. 6 is a flow diagram illustrating a shifting determination in accordance with an aspect of the present invention.

15 Fig. 7 is a flow diagram illustrating a scaling determination in accordance with an aspect of the present invention.

Fig. 8 is a schematic block diagram illustrating a suitable operating environment in accordance with an aspect of the present invention.

Fig. 9 is a schematic block diagram of a sample-computing environment with which the present invention can interact.

20

DETAILED DESCRIPTION OF THE INVENTION

The present invention relates to a system and method to facilitate data mining applications and automated evaluation of models for continuous variable data. In one aspect, a system is provided that facilitates decision tree learning. The system includes a learning component that generates non-standardized data that relates to a split in a decision tree and a scoring component that scores the split as if the non-standardized data at a subset of leaves of the decision tree had been shifted and/or scaled. In this manner, computer-processing operations can be mitigated which enhances performance of associated algorithms for evaluating the decision tree. A modification component can

25

30

also be provided for a respective candidate split score on the decision tree, wherein the above data or data subset can be modified by shifting and/or scaling the data and a new score is computed on the modified data. Furthermore, an optimization component can be provided that analyzes the data and determines whether to treat the data as if it was: (1) shifted, (2) scaled, or (3) shifted and scaled to further mitigate processing operations.

As used in this application, the terms “component,” “model,” “system,” and the like are intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a server and the server can be a component. One or more components may reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers.

Referring initially to Fig. 1, a system 100 illustrates a model evaluation system in accordance with an aspect of the present invention. Before proceeding it is noted that the present invention applies to data mining in general and decision-tree learning in particular although substantially any modeling/mining application can be similarly employed. In general, when scoring a split in a decision tree 110 (splits referring to nodes at one or more branches of the tree), the same or similar prior distribution is employed while scoring the split as if the data at respective leaves or branches of the tree 110 had been shifted and/or scaled. This mitigates the effect of the prior distribution on (1) the shape of the tree and (2) the distributions in the leaves of the tree, and thus can lead to improved out-of-sample prediction accuracy.

A learning component or model 120 is provided that inputs continuous variable data 130 (or data subset) and generates the decision tree 110 having one or more decision branches or leaves that are then scored to determine the prediction capabilities of the model. A scoring component 140 is provided that automatically analyzes the decision tree 110 in connection with various processing modes to determine respective scores for

selected branches of the tree 110. Such processing modes can be enabled by an optimization component 150 and/or modification component 160 which are described in more detail below. After the branches of the decision tree 110 have been scored, evaluation data 170 can be generated indicating how well a model predicts continuous target data and therefore whether or not the model is a suitable predictor for the target data. The evaluation data 170 can then be employed by users and/or subsequent automated components (or other entities such as local or remote networked computers) when determining model performance and/or selecting between models or model subsets.

The optimization component 150 enables the scoring component 150 to calculate scores in the above manner without actually shifting and/or scaling the data (*e.g.*, flags set to indicate shifting and/or scaling bypass operations). This enables resulting decision trees 110 to be less sensitive to “uninformative” prior information, without incurring the runtime performance penalty that may result from rescaling data each time a new split is considered or analyzed. Also, actual shifting and/or scaling operations can be applied to the data, if desired *via* the modification component 160 although system performance may be somewhat diminished.

It is to be appreciated that although continuous variable data 130 is processed in accordance with the present invention, other types of data can also be processed. For example, since continuous target variables are generally predicted by the learning component 120, input data applied to the learning models and resultant decision trees can be continuous data, non-continuous data, data subsets, and/or combinations thereof. Furthermore, in general one or more learning components or models 110 (*e.g.*, prediction models, data mining models, Bayesian models) receive data from a training data set and predict continuous variable (CV) target data. The CV target data can include substantially any type of continuous variable prediction given the training data set. As one example, given known data associated with a person (*e.g.*, education level, zip code, web sites visited, shopping selections, and so forth) predictions can be made regarding the person’s income or age which are possible examples of continuous variables. It is to be appreciated the present invention is not so limited however, in that the learning

component 110 can predict any type of continuous variable. For example, a mathematical model may observe analysis data from the training data set relating to an Engineering problem and produce a continuous variable prediction relating to one or more potential outcomes (*e.g.*, oscillatory output prediction based upon a differential equation analysis).

Referring now to Fig. 2, a system 200 illustrates an exemplary scoring component 210 in accordance with an aspect of the present invention. The scoring component 210 includes various components to automatically score splits in a decision tree. Such aspects include processing random data samples at 220, determining scoring constants at 230, determining a gamma function at 240, determining matrix and/or vector values at 250, and determining mean values for the data at 260. The following describes a generalized scoring example given basic assumptions in order to illustrate one or more aspects of the scoring component 210.

For example, assume that x_1, \dots, x_n is a random sample from a multivariate normal distribution having an unknown value of a mean vector μ and an unknown value of a covariance matrix Σ , and assume that a prior joint distribution on μ and Σ is a normal-Wishart distribution. Thus, the conditional distribution of μ given Σ is $N(\mu_0, \Sigma/\nu)$ and the marginal distribution of Σ is a Wishart distribution with α degrees of freedom and covariance matrix T_0 , is denoted $W(\alpha, T_0)$. Also, assume $\mu_0 = 0$ and T_0 is a diagonal matrix, wherein diagonal elements are equal to a pre-defined constant β (*e.g.*, $\beta = 0.01$). Assuming that there are r regressors, then a Bayesian linear regression score can be computed as:

$$score = \pi^{-n/2} \left(\frac{\nu}{\nu+n} \right)^{1/2} \frac{\Gamma(\frac{\alpha+n}{2})}{\Gamma(\frac{\alpha}{2})} \left(\beta^{\frac{\alpha+r}{2}} \right) \frac{\left(\left| \mathbf{T}_n^{\text{TR}} \right| \right)^{-\left(\frac{\alpha+n}{2} \right)}}{\left(\left| \mathbf{T}_n^{\text{R}} \right| \right)^{-\left(\frac{\alpha-1+n}{2} \right)}}, \text{ where}$$

$$\begin{aligned}
\mathbf{T}_n &= \mathbf{T}_0 + \mathbf{S}_n + \mathbf{U}_n \\
\mathbf{U}_n &= \frac{\nu n}{\nu + n} (\bar{\mu}_0 - \bar{m}_n)(\bar{\mu}_0 - \bar{m}_n)' \\
\mathbf{S}_n &= \sum_{i=1}^n (\bar{x}_i - \bar{m}_n)(\bar{x}_i - \bar{m}_n)' \\
\bar{m}_n &= \frac{1}{n} \sum_{i=1}^n \bar{x}_i
\end{aligned}$$

wherein bold-face symbols denote square matrices, symbols with overlines denote (one dimensional) vectors, the ' symbol denotes transpose, and $| |$ denotes determinant. Also, n represents a number of records in the data. Γ is a gamma function satisfying $\Gamma(x) = (x-1) \Gamma(x-1)$. A symbol \bar{x}_i denotes a vector of values for relevant variables in the i th case in the data. The superscripts TR and R in \mathbf{T}_n^{TR} and \mathbf{T}_n^{R} denote that the matrices are defined with respect to target and regressor variables in the first case and merely regressor variables in the second case. That is, when computing \mathbf{T}_n^{TR} in the numerator, the vectors and matrices have one more dimension than when computing \mathbf{T}_n^{R} in the denominator. As can be appreciated, the scoring component 210 can compute the above score in various stages and/or orderings as will be illustrated in the examples below.

Figs. 3-7 illustrate methodologies for automatically scoring and evaluating decision trees in accordance the present invention. While, for purposes of simplicity of explanation, the methodologies are shown and described as a series of acts, it is to be understood and appreciated that the present invention is not limited by the order of acts, as some acts may, in accordance with the present invention, occur in different orders and/or concurrently with other acts from that shown and described herein. For example, those skilled in the art will understand and appreciate that a methodology could alternatively be represented as a series of interrelated states or events, such as in a state diagram. Moreover, not all illustrated acts may be required to implement a methodology in accordance with the present invention. Also, it is to be appreciated that although the

following methodologies may utilize particular values for purposes of explanation, a plurality differing variable values may be employed depending on the data analyzed.

Turning to Fig. 3, a process 300 illustrates an example shifting and/or scaling determination in accordance with an aspect of the present invention. The score described above with respect to Fig. 2 can be computed as if there were shift and/or scale operations performed on the data without actually performing the operation (*e.g.*, virtual shifting and/or scaling). Alternatively, the present invention can be implemented (in a somewhat non-efficient manner) by, for each candidate split score, modifying the data by actually shifting and/or scaling the data and computing the above score on the resulting (modified) x_i , ($i = 1$ to n) data. Proceeding to 310 of Fig. 3, a determination is made as to whether to perform a virtual shifting operation. If no virtual shifting operation occurs, the process proceeds to 330. If virtual shifting is to occur, the process proceeds to 320.

At 320, if the data were shifted, it is known that all of the $(\bar{\mu}_0 - \bar{m}_n)$ terms described above will be zero, and hence the U_n term from the scoring component in the computation of T_n cancels. The matrix S obtains the same value, regardless of whether data is shifted or not, so it can be computed (as if data had shifted) without any explicit shift. Proceeding to 330, another determination is made as to whether to apply virtual scaling. If not, the process proceeds to 340 and computes a score with the shifting options if selected at 310. If scaling is selected at 330, the process proceeds to 340.

At 340, if the data were scaled, all elements of the matrix S_n (or element subset) described above are adjusted. In particular, the (j,k) element of matrix S_n are modified by dividing by the product of the sample standard deviations of each of the two variables. The sample standard deviation can be computed *via*:

$$SD(x^j) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i^j - m_n^j)^2}$$

wherein x_i^j denotes the value of the j th variable in the i th case of the data, and m_n^j denotes the sample average for the j th variable. (Those of ordinary skill in the art will appreciate that there are alternative formula for the sample standard deviation.)

Referring now to Fig. 4, a process 400 illustrates an example of intermediate scoring determinations in accordance with an aspect of the present invention. Proceeding to 420, a model form is selected or determined. For example, assume there are two variables T and R, where T is a target variable and R is a regressor variable. In other words, the model is a linear regression of the form:

$$T = a * R + b$$

Also, assume that the following data is at a particular leaf in a decision tree, and then calculate the score of that leaf using the *score* function described above with respect to Fig. 2.

	x^T	x^R
Case 1	2	4
Case 2	3	0
Case 3	4	-4

At 430, scoring constants are determined. In this case, the number of regressors r is 1, and the number of cases n is 3. In general, $\beta = 0.01$ (or other value) and $\nu = 1$ and thus, $\alpha = \nu + r + 1 = 3$.

To compute the score, it is useful to collect the following statistics from the data by determining a mean computation at 440. Sample means of T and R, can be expressed as a vector (this vector is identical to \bar{m}_n for computing the numerator of the score):

$$\text{SampMean}^{\text{TR}} = \left(\frac{1}{n} \sum_{i=1}^n x^T_i, \frac{1}{n} \sum_{i=1}^n x^R_i \right) = \left(\frac{1}{3} (2 + 3 + 4), \frac{1}{3} (4 + 0 - 4) \right) = (3, 0)$$

At 450, a sample variance/covariance matrix is determined as follows:

$$\begin{aligned}
\mathbf{SampCov}^{\text{TR}} &= \begin{pmatrix} \frac{1}{n} \sum_{i=1}^n (\bar{x}_i^T - \bar{m}_n^T) \times (\bar{x}_i^T - \bar{m}_n^T) & \frac{1}{n} \sum_{i=1}^n (\bar{x}_i^T - \bar{m}_n^T) \times (\bar{x}_i^R - \bar{m}_n^R) \\ \frac{1}{n} \sum_{i=1}^n (\bar{x}_i^R - \bar{m}_n^R) \times (\bar{x}_i^T - \bar{m}_n^T) & \frac{1}{n} \sum_{i=1}^n (\bar{x}_i^R - \bar{m}_n^R) \times (\bar{x}_i^R - \bar{m}_n^R) \end{pmatrix} = \\
&\begin{pmatrix} \frac{1}{3} [(2-3)(2-3) + (3-3)(3-3) + (4-3)(4-3)] & \frac{1}{3} [(2-3)(4-0) + (3-3)(0-0) + (4-3)(-4-0)] \\ \frac{1}{3} [(4-0)(2-3) + (0-0)(3-3) + (-4-0)(4-3)] & \frac{1}{3} [(4-0)(4-0) + (0-0)(0-0) + (-4-0)(-4-0)] \end{pmatrix} = \\
&\begin{pmatrix} \frac{2}{3} & -\frac{8}{3} \\ -\frac{8}{3} & \frac{32}{3} \end{pmatrix}
\end{aligned}$$

Those of ordinary skill in the art will appreciate that there are alternative formula for the sample covariance. Also, as noted above, the particular numbers employed in these examples are for illustrative purposes and other values can be selected. From the sample mean and sample covariance matrix, the appropriate vectors can be extracted in the computation for score.

Fig. 5 illustrates an example scoring process that is a continuation of the process described with respect to Fig. 4. At 510, assuming no shifting and/or scaling is selected, numerator vectors and matrices are determined as follows:

A. The numerator \mathbf{T}_n^{TR}

In this case, the vectors and matrices are defined for x^T and x^R :

$$\begin{aligned}
\mathbf{T}_0^{\text{TR}} &= \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix} \\
\mathbf{U}_n^{\text{TR}} &= \frac{1 \times 3}{1+3} \begin{bmatrix} 0-3 \\ 0-0 \end{bmatrix} \begin{bmatrix} 0-3 & 0-0 \end{bmatrix} = \frac{3}{4} \begin{pmatrix} 9 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} \frac{27}{4} & 0 \\ 0 & 0 \end{pmatrix} =
\end{aligned}$$

$$\mathbf{S}_n^{\text{TR}} = n \times \mathbf{SampCov}^{\text{TR}} = \begin{pmatrix} 2 & -8 \\ -8 & 32 \end{pmatrix}$$

and therefore

$$\mathbf{T}_n^{\text{TR}} = \begin{pmatrix} \frac{876}{100} & -8 \\ -8 & \frac{3201}{100} \end{pmatrix}$$

and

$$|\mathbf{T}_n^{\text{TR}}| = 216.4076$$

5

At 520, denominator vector and matrices are determined.

10 B. The denominator \mathbf{T}_n^{R} is computed almost the same as above, but instead of having 2x2 dimensional matrices and 2 dimensional vectors over both the target and regressor, 1x1 dimensional matrices and 1 dimensional vectors are applied over the regressor. In general, the T matrix for the denominator can be extracted directly from the T matrix for the numerator by pulling out the row and column corresponding to the target. In this example, this includes removing the top row and left-most column, leaving:

$$\mathbf{T}_n^{\text{R}} = \left(\frac{3201}{100} \right)$$

15

and

$$|\mathbf{T}_n^{\text{R}}| = 32.01$$

Given these matrices then the score can be computed at 530 as:

20

$$score = \pi^{-3/2} \left(\frac{1}{1+3} \right)^{1/2} \frac{\Gamma\left(\frac{3+3}{2}\right)}{\Gamma\left(\frac{3}{2}\right)} \left(0.01^{\frac{3+1}{2}} \right) \frac{(216.4076)^{-\left(\frac{3+3}{2}\right)}}{(32.01)^{-\left(\frac{3-1+3}{2}\right)}}$$

Figs. 6 and 7 illustrate example scaling and/or scoring optimizations in accordance with an aspect of the present invention. Proceeding to 610 of Fig. 6, a determination is made as to whether to apply the shifting optimization. If not, the mean is subtracted from each case in the data at 620 and the scoring is determined as described

with respect to Fig. 2. If the shifting optimization is desired at 610, the U_n term can be omitted at 630 when calculating T_n before determining the score at 640. For example:

$$T_n^{TR} \text{ shifted} = T_0^{TR} + S_n^{TR} = \begin{pmatrix} \frac{201}{100} & -8 \\ -8 & \frac{3201}{100} \end{pmatrix}$$

5 A similar determination for scaling is illustrated at Fig. 7. For example, if scaling optimization is not selected at 710, then divide the value of each variable in each case by the sample standard deviation of that variable and determine the score at 740 as described above with respect to Fig. 2. If scaling optimization is selected at 710, divide each
10 element in S_n by the corresponding sample standard deviations as described above. It is noted that the sample standard deviation for a variable is the square root corresponding diagonal element in the sample variance/covariance matrix SampCov^{TR} . Thus:

$$S_n^{TR} \text{ scaled} = \begin{pmatrix} \frac{2}{\sqrt{\frac{2}{3}} \times \sqrt{\frac{2}{3}}} & \frac{-8}{\sqrt{\frac{2}{3}} \times \sqrt{\frac{32}{3}}} \\ \frac{-8}{\sqrt{\frac{32}{3}} \times \sqrt{\frac{2}{3}}} & \frac{32}{\sqrt{\frac{32}{3}} \times \sqrt{\frac{32}{3}}} \end{pmatrix}$$

and

$$15 \quad T_n^{TR} \text{ scaled} = T_0^{TR} + S_n^{TR} \text{ scaled} + U_n^{TR}$$

As can be appreciated, both shifting and scaling optimizations can be selected. Thus, the data can be adjusted by first subtracting the mean and then dividing by the standard deviation, or apply both shifting and scaling optimizations as follows:

$$20 \quad T_n^{TR} \text{ shifted/scaled} = T_0^{TR} + S_n^{TR} \text{ scaled}$$

With reference to Fig. 8, an exemplary environment 810 for implementing various aspects of the invention includes a computer 812. The computer 812 includes a processing unit 814, a system memory 816, and a system bus 818. The system bus 818 couples system components including, but not limited to, the system memory 816 to the processing unit 814. The processing unit 814 can be any of various available processors. Dual microprocessors and other multiprocessor architectures also can be employed as the processing unit 814.

The system bus 818 can be any of several types of bus structure(s) including the memory bus or memory controller, a peripheral bus or external bus, and/or a local bus using any variety of available bus architectures including, but not limited to, 11-bit bus, Industrial Standard Architecture (ISA), Micro-Channel Architecture (MSA), Extended ISA (EISA), Intelligent Drive Electronics (IDE), VESA Local Bus (VLB), Peripheral Component Interconnect (PCI), Universal Serial Bus (USB), Advanced Graphics Port (AGP), Personal Computer Memory Card International Association bus (PCMCIA), and Small Computer Systems Interface (SCSI).

The system memory 816 includes volatile memory 820 and nonvolatile memory 822. The basic input/output system (BIOS), containing the basic routines to transfer information between elements within the computer 812, such as during start-up, is stored in nonvolatile memory 822. By way of illustration, and not limitation, nonvolatile memory 822 can include read only memory (ROM), programmable ROM (PROM), electrically programmable ROM (EPROM), electrically erasable ROM (EEPROM), or flash memory. Volatile memory 820 includes random access memory (RAM), which acts as external cache memory. By way of illustration and not limitation, RAM is available in many forms such as synchronous RAM (SRAM), dynamic RAM (DRAM), synchronous DRAM (SDRAM), double data rate SDRAM (DDR SDRAM), enhanced SDRAM (ESDRAM), Synchlink DRAM (SLDRAM), and direct Rambus RAM (DRRAM).

Computer 812 also includes removable/non-removable, volatile/non-volatile computer storage media. Fig. 8 illustrates, for example a disk storage 824. Disk storage

824 includes, but is not limited to, devices like a magnetic disk drive, floppy disk drive, tape drive, Jaz drive, Zip drive, LS-100 drive, flash memory card, or memory stick. In addition, disk storage 824 can include storage media separately or in combination with other storage media including, but not limited to, an optical disk drive such as a compact disk ROM device (CD-ROM), CD recordable drive (CD-R Drive), CD rewritable drive (CD-RW Drive) or a digital versatile disk ROM drive (DVD-ROM). To facilitate connection of the disk storage devices 824 to the system bus 818, a removable or non-removable interface is typically used such as interface 826.

It is to be appreciated that Fig 8 describes software that acts as an intermediary between users and the basic computer resources described in suitable operating environment 810. Such software includes an operating system 828. Operating system 828, which can be stored on disk storage 824, acts to control and allocate resources of the computer system 812. System applications 830 take advantage of the management of resources by operating system 828 through program modules 832 and program data 834 stored either in system memory 816 or on disk storage 824. It is to be appreciated that the present invention can be implemented with various operating systems or combinations of operating systems.

A user enters commands or information into the computer 812 through input device(s) 836. Input devices 836 include, but are not limited to, a pointing device such as a mouse, trackball, stylus, touch pad, keyboard, microphone, joystick, game pad, satellite dish, scanner, TV tuner card, digital camera, digital video camera, web camera, and the like. These and other input devices connect to the processing unit 814 through the system bus 818 *via* interface port(s) 838. Interface port(s) 838 include, for example, a serial port, a parallel port, a game port, and a universal serial bus (USB). Output device(s) 840 use some of the same type of ports as input device(s) 836. Thus, for example, a USB port may be used to provide input to computer 812, and to output information from computer 812 to an output device 840. Output adapter 842 is provided to illustrate that there are some output devices 840 like monitors, speakers, and printers, among other output devices 840, that require special adapters. The output adapters 842 include, by way of

illustration and not limitation, video and sound cards that provide a means of connection between the output device 840 and the system bus 818. It should be noted that other devices and/or systems of devices provide both input and output capabilities such as remote computer(s) 844.

5 Computer 812 can operate in a networked environment using logical connections to one or more remote computers, such as remote computer(s) 844. The remote computer(s) 844 can be a personal computer, a server, a router, a network PC, a workstation, a microprocessor based appliance, a peer device or other common network node and the like, and typically includes many or all of the elements described relative to
10 computer 812. For purposes of brevity, only a memory storage device 846 is illustrated with remote computer(s) 844. Remote computer(s) 844 is logically connected to computer 812 through a network interface 848 and then physically connected *via* communication connection 850. Network interface 848 encompasses communication networks such as local-area networks (LAN) and wide-area networks (WAN). LAN
15 technologies include Fiber Distributed Data Interface (FDDI), Copper Distributed Data Interface (CDDI), Ethernet/IEEE 1102.3, Token Ring/IEEE 1102.5 and the like. WAN technologies include, but are not limited to, point-to-point links, circuit switching networks like Integrated Services Digital Networks (ISDN) and variations thereon, packet switching networks, and Digital Subscriber Lines (DSL).

20 Communication connection(s) 850 refers to the hardware/software employed to connect the network interface 848 to the bus 818. While communication connection 850 is shown for illustrative clarity inside computer 812, it can also be external to computer 812. The hardware/software necessary for connection to the network interface 848 includes, for exemplary purposes only, internal and external technologies such as,
25 modems including regular telephone grade modems, cable modems and DSL modems, ISDN adapters, and Ethernet cards.

Fig. 9 is a schematic block diagram of a sample-computing environment 900 with which the present invention can interact. The system 900 includes one or more client(s) 910. The client(s) 910 can be hardware and/or software (*e.g.*, threads, processes,

computing devices). The system 900 also includes one or more server(s) 930. The server(s) 930 can also be hardware and/or software (*e.g.*, threads, processes, computing devices). The servers 930 can house threads to perform transformations by employing the present invention, for example. One possible communication between a client 910 and a server 930 may be in the form of a data packet adapted to be transmitted between two or more computer processes. The system 900 includes a communication framework 950 that can be employed to facilitate communications between the client(s) 910 and the server(s) 930. The client(s) 910 are operably connected to one or more client data store(s) 960 that can be employed to store information local to the client(s) 910. Similarly, the server(s) 930 are operably connected to one or more server data store(s) 940 that can be employed to store information local to the servers 930.

What has been described above includes examples of the present invention. It is, of course, not possible to describe every conceivable combination of components or methodologies for purposes of describing the present invention, but one of ordinary skill in the art may recognize that many further combinations and permutations of the present invention are possible. Accordingly, the present invention is intended to embrace all such alterations, modifications and variations that fall within the spirit and scope of the appended claims. Furthermore, to the extent that the term “includes” is used in either the detailed description or the claims, such term is intended to be inclusive in a manner similar to the term “comprising” as “comprising” is interpreted when employed as a transitional word in a claim.